#WWDC18

Integrating with Photos on macOS

Session 505

Eric Hanson, Technology Evangelist
Tobias Conradi, Photos Engineer
Sanaa Squalli, Photos Engineer
Joachim Fornallaz, Photos Engineer

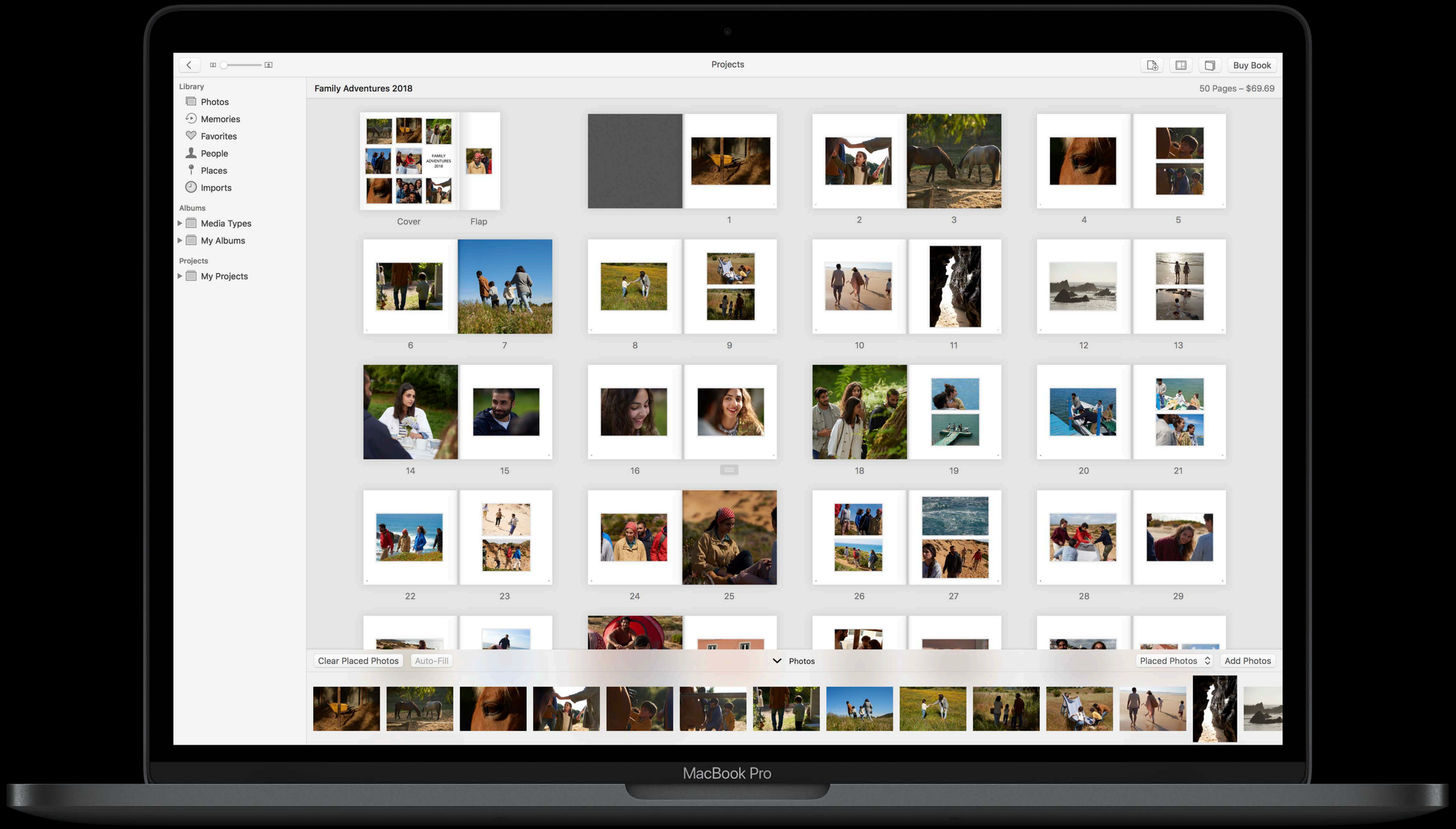
© 2018 Apple Inc. All rights reserved. Redistribution or public display not permitted without written permission from Apple.

Update on Photos Project Extensions

Interaction between applications

Update on Photos Project Extensions

Interaction between applications

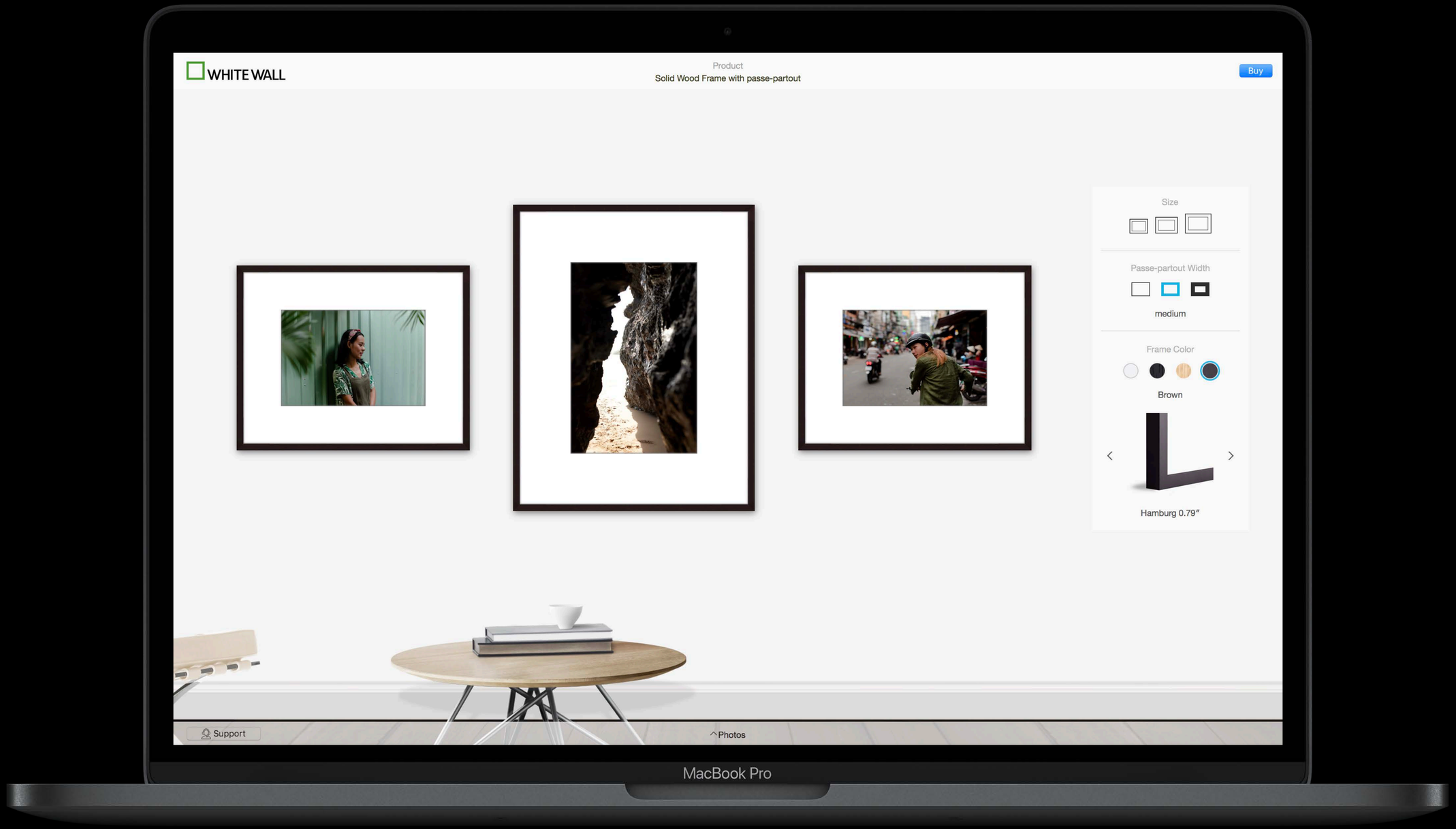

Projects
Buy Book
Library
Photos
Memories
Favorites
People
Places
Imports
Albums
Media Types
My Albums
Projects
My Projects
Family Adventures 2018
50 Pages – $69.69
Cover
Flap
Clear Placed Photos
Auto-Fill
Photos
Placed Photos
Add Photos
MacBook Pro

70 million

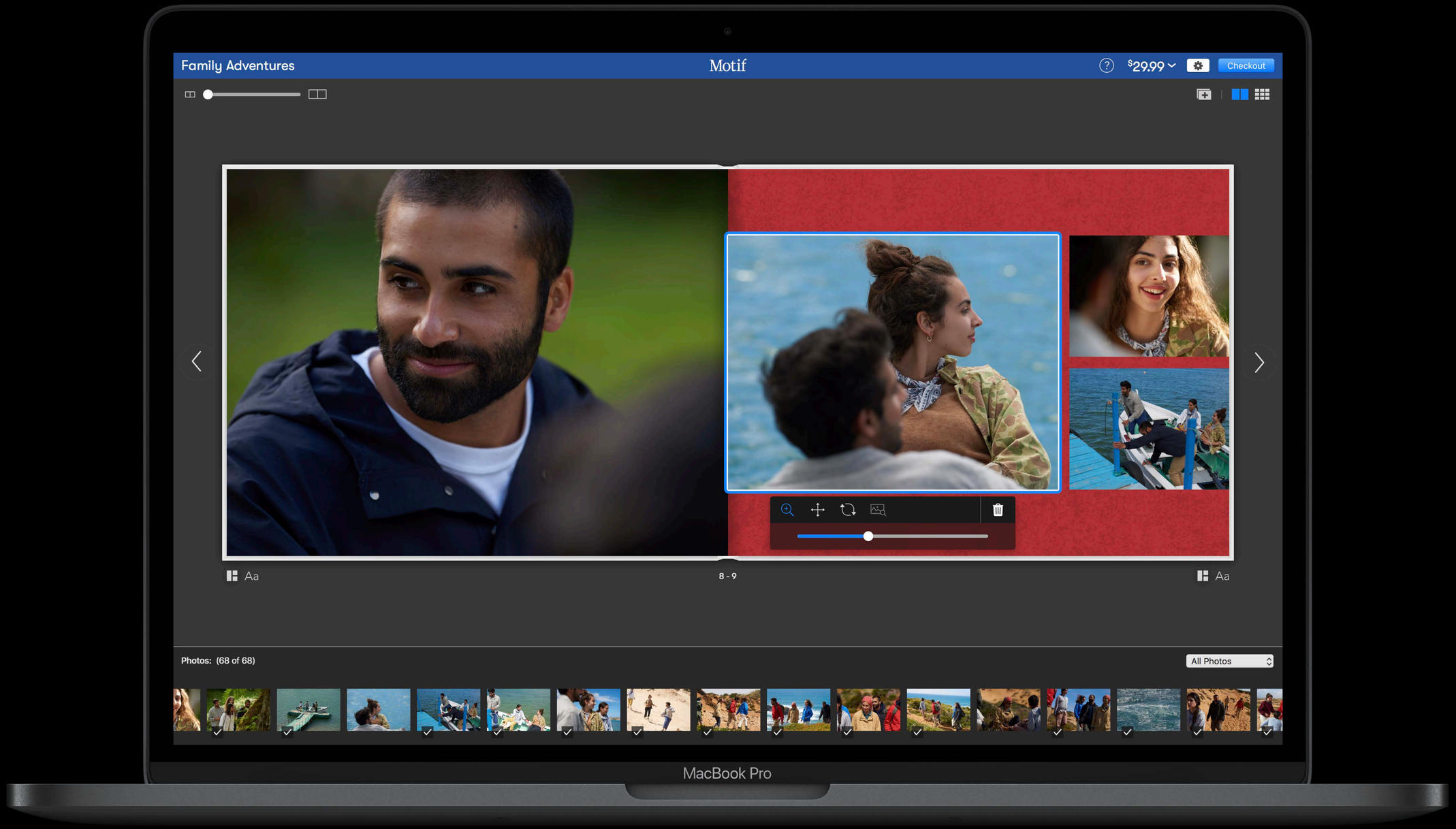
Books, calendars, and cards created in iPhoto and Photos

T-Shirts
Announcements
Photo Boxes
Posters
Banners
Magnets
Comic Books
Wall Decor
Comic Books
Wedding Albums
Personal Websites
Digital Books
iPhone Cases
Key Chains
Pillows
Photo Rugs
Photo Books
Stickers
Web Galleries
Record Sleeves
Desk Calendars
Collages
Greeting Cards
Puzzles
Backlit Displays
Mobiles
Coasters
Framed Prints
Postcards
Gallery Walls
Photo Cubes
Metal Prints
Wall Calendars
Home Decor
Slideshows
Yearbooks
Wood Prints
Brochures
Invitations
Prints
Photo Gifts
Cookbooks
Photo Journals
Coloring Books
Ornaments
MacBook Skins
Flyers
Interactive Stories
Marshmallows
Travel Journals
Laser Engraving
CD Packaging
Blankets
Photo Blogs
Canvas Prints
Coffee Mugs

Change Product
Change Theme
10 x 10 Hard Cover Book
Buy $51.87
PHOTO
TEXT
LAYOUT
BACKGROUND
mimeo
PHOTOS
Organize
2-3
Front Cover
MacBook Pro


WHITE WALL
Product
Solid Wood Frame with passe-partout
Buy
Size
Passe-partout Width
medium
Frame Color
Brown
Hamburg 0.79″
Support
Photos
MacBook Pro

Help
WIX PhotoAlbums
All changes saved
Publish Album
Family Adventures
June 7, 2018
View Photos
MacBook Pro


Family Adventures
Motif
$29.99
Checkout
Aa
8 - 9
Aa
Photos: (68 of 68)
All Photos
MacBook Pro

ifolor.

mimeo Photos

Motif

WHITE WALL

Shutterfly®

mpix

FUJIFILM

WIX.com

New in Photos Project Extensions

Photos File Edit Image View Window Help

Tue 9:41 AM

Photos - ifolor Photo Products

Library
- Photos
- Memories
- Favorites
- People
- Places
- Imports
- Recently Deleted

Albums
- Media Types
- My Albums
  - Family Adventures
  - Great Shots
  - Edit Examples
  - Our Family
  - At Home
  - Coast Trip
  - Emily's 10th Birthday
  - Southwest
  - Exploring Marin
  - Hiking Russian Ridge
  - Beautiful China
  - Santa Cruz Beach B...
  - Berry Farm
  - Horseback Riding
  - Lighthouse
  - Surfing
  - Mountain Biking
  - Pumpkin Picking
  - San Francisco
  - Guitar Lesson
  - Summer in Tahoe
  - In the Garden
  - Tomales Bay
  - Winter in Tahoe
  - 4th of July BBQ
  - HDR Shots
  - Edited Shots

Projects
- My Projects

ifolor. Gallery Print, 30×40 cm, including wall mounting

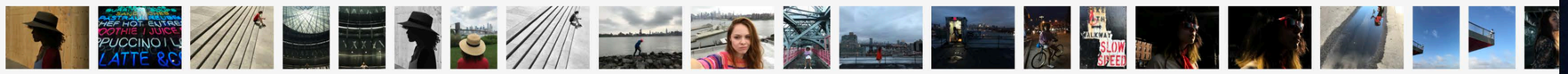
Order

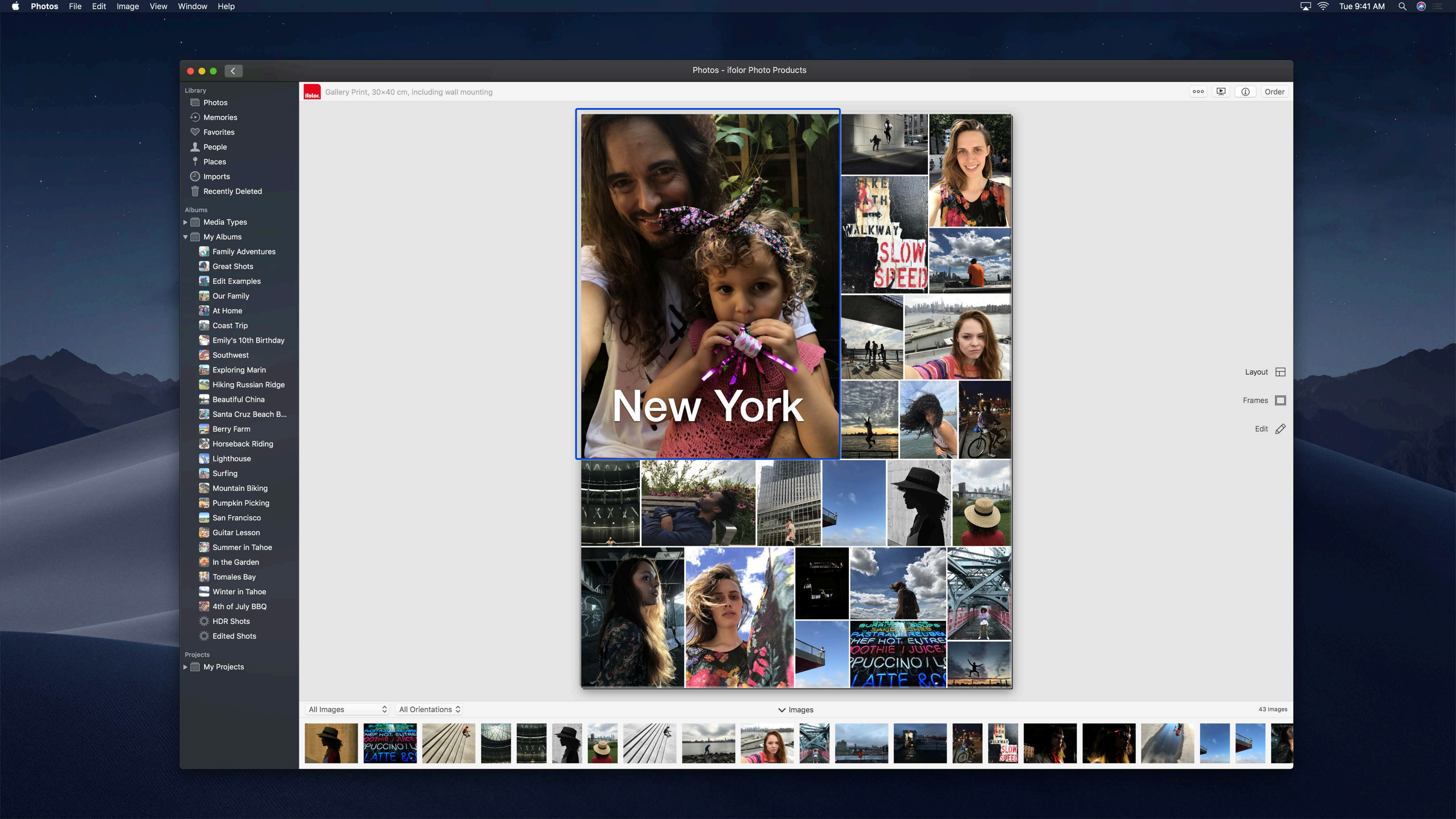
Layout

Frames

Edit

All Images | All Orientations | Images | 43 Images




Photos File Edit Image View Window Help
Tue 9:41 AM
Photos - ifolor Photo Products
Library
Photos
Memories
Favorites
People
Places
Imports
Recently Deleted
Albums
Media Types
My Albums
Family Adventures
Great Shots
Edit Examples
Our Family
At Home
Coast Trip
Emily's 10th Birthday
Southwest
Exploring Marin
Hiking Russian Ridge
Beautiful China
Santa Cruz Beach B...
Berry Farm
Horseback Riding
Lighthouse
Surfing
Mountain Biking
Pumpkin Picking
San Francisco
Guitar Lesson
Summer in Tahoe
In the Garden
Tomales Bay
Winter in Tahoe
4th of July BBQ
HDR Shots
Edited Shots
Projects
My Projects
Gallery Print, 30×40 cm, including wall mounting
Order
New York
Layout
Frames
Edit
All Images
All Orientations
Images
43 Images

Photos File Edit Image View Window Help

Tue 9:41 AM

Revert to Original

Adjust | Filters | Crop

Done

ADJUST

- Light
  - Options
- Color
- Black & White
  - Options
- Retouch
- Red-eye
- White Balance
- Curves
- Levels
- Definition
  - Amount 0.42
- Selective Color
  - Hue 0.00
  - Saturation 0.00
  - Luminance 0.00
  - Range 1.00
- Noise Reduction
  - Amount 0.00
- Sharpen
  - Intensity 0.11
  - Edges 0.22
  - Falloff 0.69
- Vignette
  - Strength 0.57
  - Radius 0.50
  - Softness 0.00

Live

Reset Adjustments

Photos File Edit Image View Window Help

Tue 9:41 AM

Photos - ifolor Photo Products

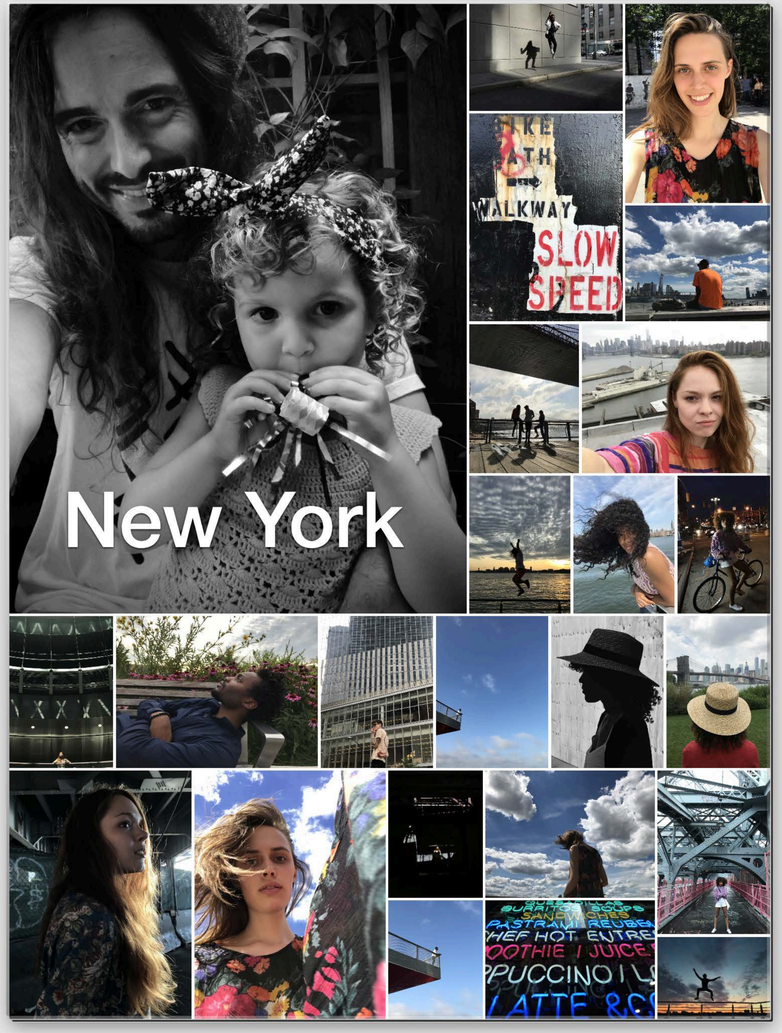
Library
- Photos
- Memories
- Favorites
- People
- Places
- Imports
- Recently Deleted

Albums
- Media Types
- My Albums
  - Family Adventures
  - Great Shots
  - Edit Examples
  - Our Family
  - At Home
  - Coast Trip
  - Emily's 10th Birthday
  - Southwest
  - Exploring Marin
  - Hiking Russian Ridge
  - Beautiful China
  - Santa Cruz Beach B...
  - Berry Farm
  - Horseback Riding
  - Lighthouse
  - Surfing
  - Mountain Biking
  - Pumpkin Picking
  - San Francisco
  - Guitar Lesson
  - Summer in Tahoe
  - In the Garden
  - Tomales Bay
  - Winter in Tahoe
  - 4th of July BBQ
  - HDR Shots
  - Edited Shots

Projects
- My Projects

ifolor. Gallery Print, 30×40 cm, including wall mounting

Order

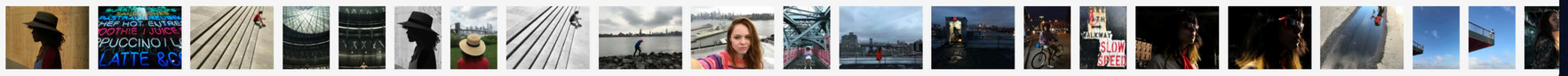


Layout

Frames

Edit

All Images | All Orientations | Images | 43 Images



Photos File Edit Image View Window Help

Tue 9:41 AM

My Albums

Search

Library

- Photos
- Memories
- Favorites
- People
- Places
- Imports

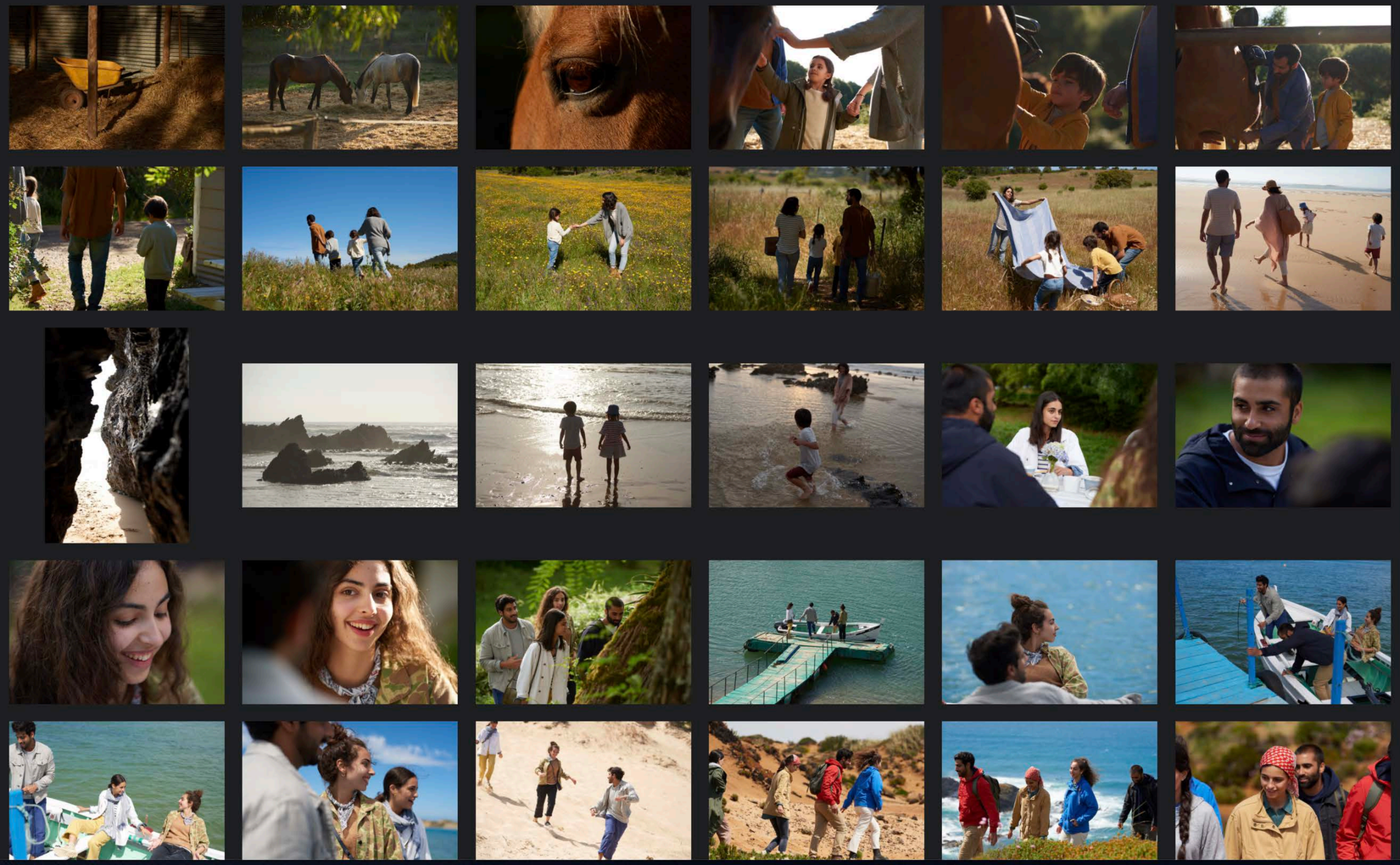
Albums

- Media Types
- My Albums
  - Family Adventur...
  - Great Shots
  - Edit Examples
  - Our Family
  - At Home
  - Berry Farm
  - Coast Trip
  - Emily's 10th Birt...
  - Southwest
  - Exploring Marin
  - Hiking Russian...
  - Beautiful China
  - Santa Cruz Bea...
  - Horseback Riding
  - Lighthouse
  - Surfing
  - Mountain Biking
  - Pumpkin Picking
  - San Francisco
  - Guitar Lesson
  - Summer in Tahoe
  - In the Garden
  - Tomales Bay
  - Winter in Tahoe

Family Adventures

May – June 2017 68 Photos Show as Memory Slideshow

Showing: All Photos



App Store File Edit Store Window Help

Tue 9:41 AM

Done

GET STARTED

Create Projects With Photos

Photos for macOS is not only useful for collecting, editing, and organizing your pictures—it's a great destination for creating print projects out of your very best shots. While you can start with books, calendars, cards, slide shows, or prints, your options expand with Photos project extensions from a variety of apps.

Mimeo Photos

Design and lay out customized photo books, calendars, and cards.

Mimeo Photos
Photography

GET

WhiteWall

Turn photos into gallery-quality wall art.

WhiteWall - for
Photography

GET

Wix

Create and share beautiful online photo albums.

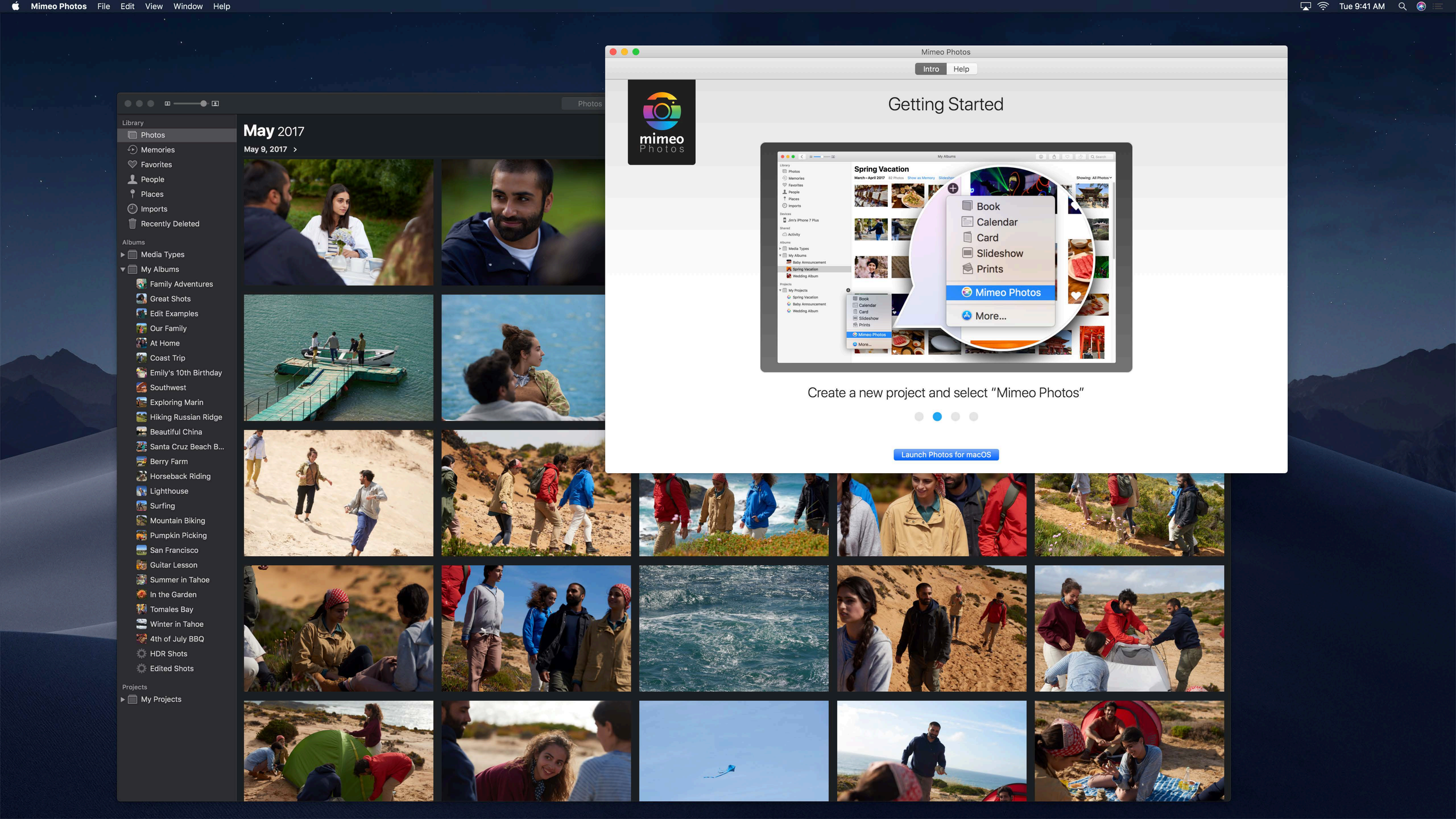
Wix Photo
Photography

Library
- Photos
- Memories
- Favorites
- People
- Places
- Imports

Albums
- Media Types
- My Albums
  - Family Adventur...
  - Great Shots
  - Edit Examples
  - Our Family
  - At Home
  - Berry Farm
  - Coast Trip
  - Emily's 10th Birt...
  - Southwest
  - Exploring Marin
  - Hiking Russian...
  - Beautiful China
  - Santa Cruz Bea...
  - Horseback Riding
  - Lighthouse
  - Surfing
  - Mountain Biking
  - Pumpkin Picking
  - San Francisco
  - Guitar Lesson
  - Summer in Tahoe
  - In the Garden
  - Tomales Bay
  - Winter in Tahoe

Search

Showing: All Photos


Mimeo Photos File Edit View Window Help
Tue 9:41 AM
Photos
Library
Photos
Memories
Favorites
People
Places
Imports
Recently Deleted
Albums
Media Types
My Albums
Family Adventures
Great Shots
Edit Examples
Our Family
At Home
Coast Trip
Emily's 10th Birthday
Southwest
Exploring Marin
Hiking Russian Ridge
Beautiful China
Santa Cruz Beach B...
Berry Farm
Horseback Riding
Lighthouse
Surfing
Mountain Biking
Pumpkin Picking
San Francisco
Guitar Lesson
Summer in Tahoe
In the Garden
Tomales Bay
Winter in Tahoe
4th of July BBQ
HDR Shots
Edited Shots
Projects
My Projects
May 2017
May 9, 2017
Mimeo Photos
Intro
Help
mimeo Photos
Getting Started
Spring Vacation
Book
Calendar
Card
Slideshow
Prints
Mimeo Photos
More...
Create a new project and select "Mimeo Photos"
Launch Photos for macOS

NEW

```
// Launching Photos to create a project

func applicationDidFinishLaunching(_ aNotification: Notification) {

        let urlComponents = NSURLComponents()
        urlComponents.scheme = "photos"
        urlComponents.path = "projectExtension/createProject"

        // specify extension and optional category
        let extensionQuery =

                URLQueryItem(name: "extensionIdentifier", value: myExtensionIdentifier)

        let categoryQuery =

                URLQueryItem(name: "category", value: PHProjectCategory.walldecor.rawValue)

        urlComponents.queryItems = [extensionQuery, categoryQuery]

        NSWorkspace.shared.open(urlComponents.url!)

}
```

NEW

```
// Launching Photos to create a project

func applicationDidFinishLaunching(_ aNotification: Notification) {

        let urlComponents = NSURLComponents()
        urlComponents.scheme = "photos"
        urlComponents.path = "projectExtension/createProject"

        // specify extension and optional category
        let extensionQuery =

                URLQueryItem(name: "extensionIdentifier", value: myExtensionIdentifier)

        let categoryQuery =

                URLQueryItem(name: "category", value: PHProjectCategory.walldecor.rawValue)

        urlComponents.queryItems = [extensionQuery, categoryQuery]

        NSWorkspace.shared.open(urlComponents.url!)

}
```

NEW

```
// Launching Photos to create a project

func applicationDidFinishLaunching(_ aNotification: Notification) {

        let urlComponents = NSURLComponents()
        urlComponents.scheme = "photos"
        urlComponents.path = "projectExtension/createProject"

        // specify extension and optional category
        let extensionQuery =

                URLQueryItem(name: "extensionIdentifier", value: myExtensionIdentifier)

        let categoryQuery =

                URLQueryItem(name: "category", value: PHProjectCategory.walldecor.rawValue)

        urlComponents.queryItems = [extensionQuery, categoryQuery]

        NSWorkspace.shared.open(urlComponents.url!)

}
```

Photos File Edit Image View Window Help

Tue 9:41 AM

Photos | Moments | Collections | Years

Search

Library
- Photos
- Memories
- Favorites
- People
- Places
- Imports

Albums
- Media Types
- My Albums

Projects
- My Projects

May 2017

May 9, 2017 >

Showing: All Photos

Mimeo Photos

Photobooks

Tell your personalized story, one volume at a time.

Crafted and shipped within 3 days. Starting at $9.99.

Calendars

Brighten each month with favorite selections from your photo collections.

Crafted and shipped within 3 days. $19.99 for 12 months.

Cards

Create personalized correspondence for those you care about.

Crafted and shipped within 3 days. Starting at $0.99.

Cancel | Next

Photos File Edit Image View Window Help

Tue 9:41 AM

Projects

Search

Library
- Photos
- Memories
- Favorites
- People
- Places
- Imports

Albums
- Media Types
- My Albums

Projects
- My Projects

Santa Cruz Beach Boardwalk
XL Hardcover Book

On the Road
XL Hardcover Book

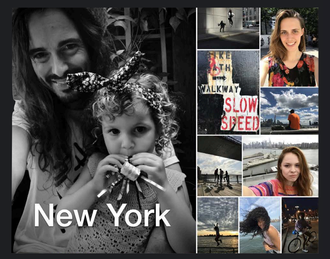
Utah Trip
XL Hardcover Book

Family Ski Trip
Slideshow

Journey to Iceland
Slideshow

Summer in Tahoe
XL Hardcover Book

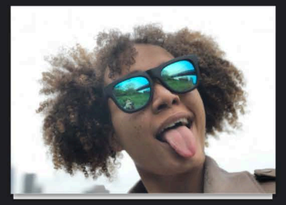
Photo Mug for Mom



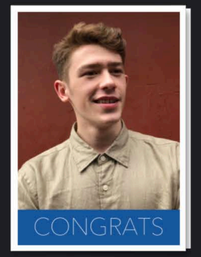
New York Collage Frame

Winter in Tahoe
L Hardcover Book

Family Adventures 2018
XL Hardcover Book



Happy Birthday!
Folded Card



Brooklyn, NY
Folded Card

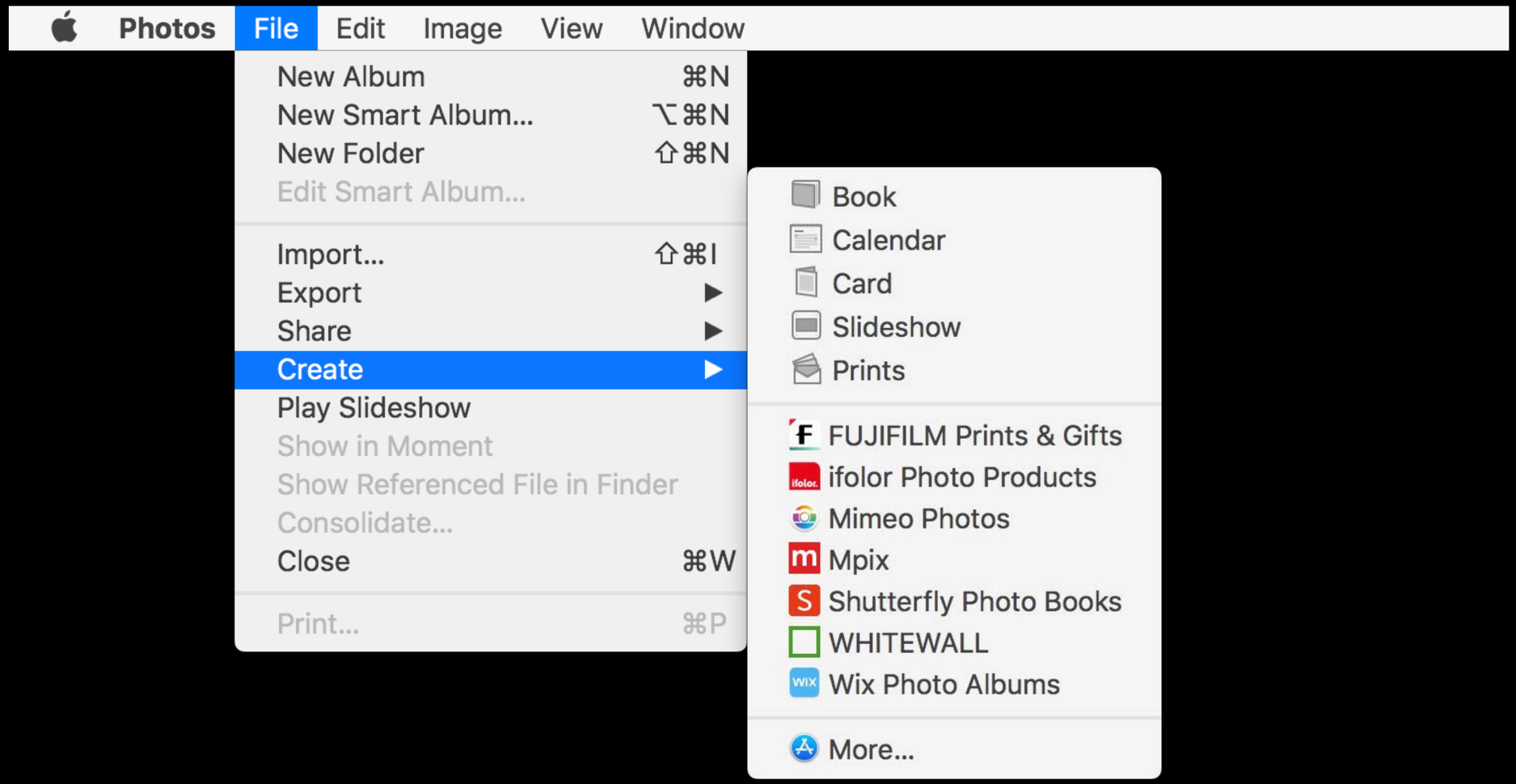
Photos Project Extensions

A closer look at everything new

Tobias Conradi, Photos Engineer

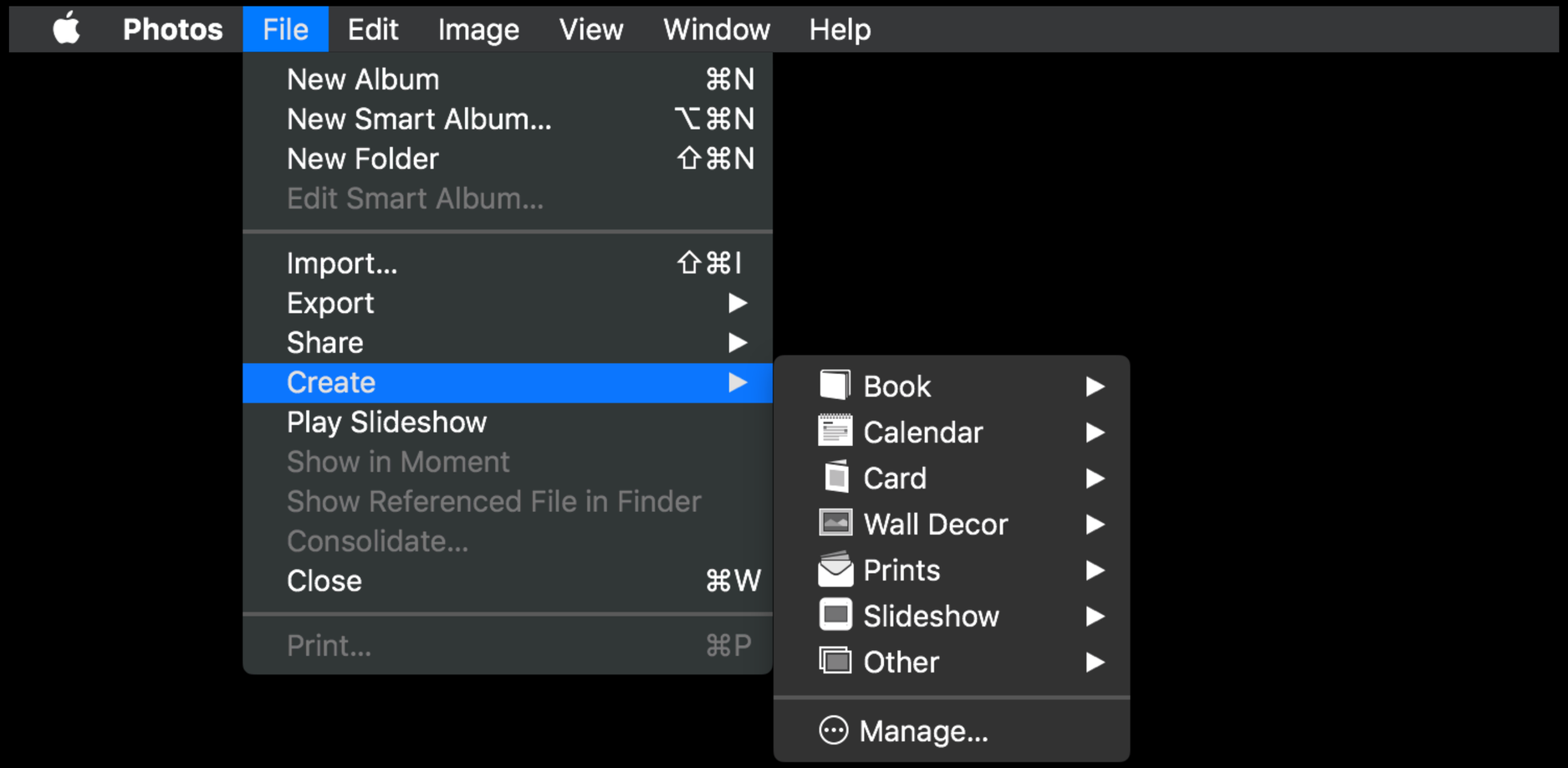
Project Categories

High Sierra



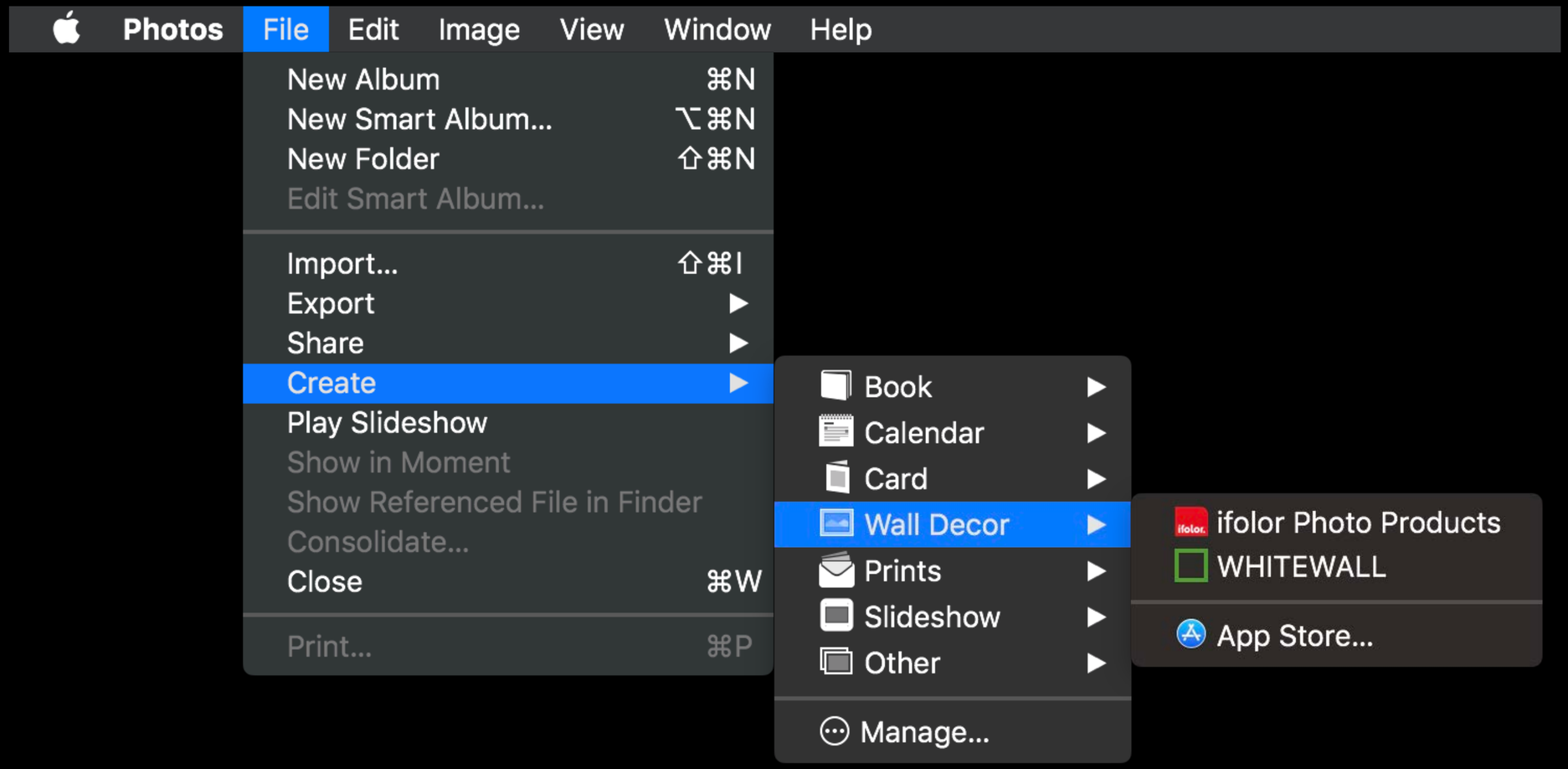
Project Categories

Mojave



Project Categories

Mojave



```
//Project Categories

/<!— Excerpt from an Extensions Info.plist —->
<key>NSExtension</key>
<dict>
    <key>NSExtensionAttributes</key>
    <dict>
        <key>PHProjectCategory</key>
        <array>
            <string>walldecor</string>
            <string>other</string>
        </array>
    </dict>
    <!-- ... -->
```

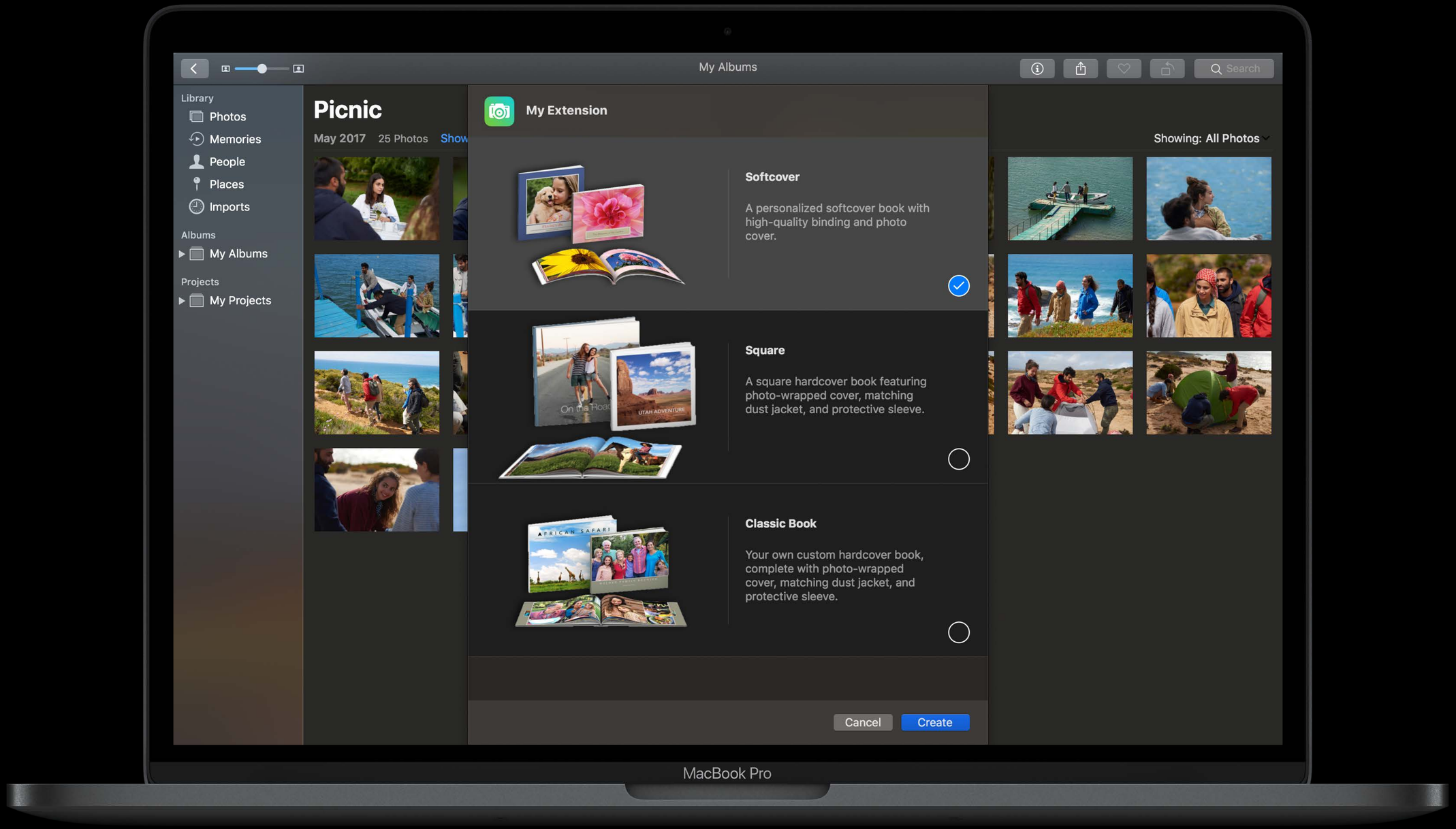
NEW

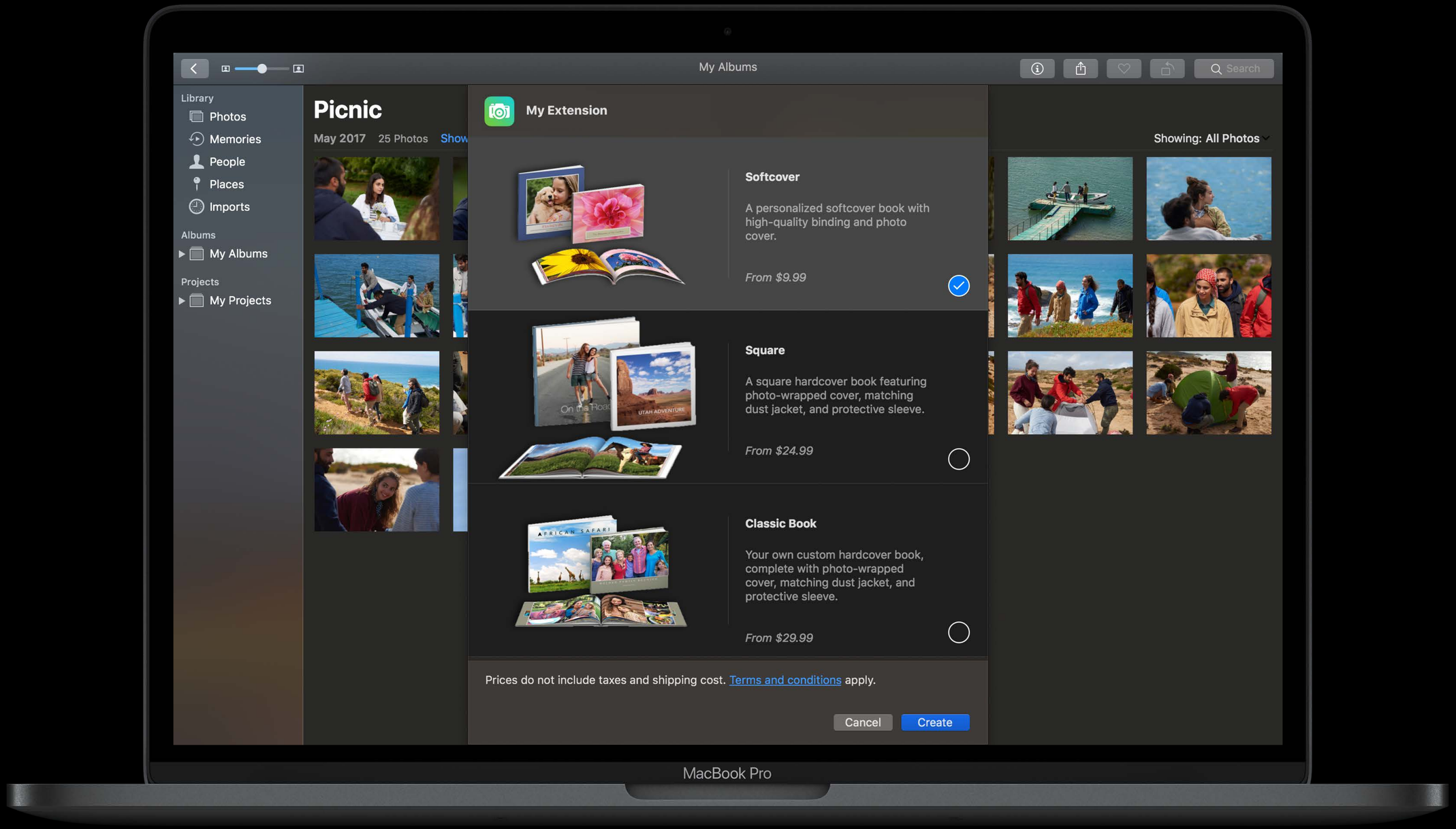
```
//Project Categories

/<!— Excerpt from an Extensions Info.plist —->
<key>NSExtension</key>
<dict>
    <key>NSExtensionAttributes</key>
    <dict>
        <key>PHProjectCategory</key>
        <array>
            <string>walldecor</string>
            <string>other</string>
        </array>
    </dict>
    <!-- ... -->
```

NEW

```
//Project Categories

/<!— Excerpt from an Extensions Info.plist —->
<key>NSExtension</key>
<dict>
    <key>NSExtensionAttributes</key>
    <dict>
        <key>PHProjectCategory</key>
        <array>
            <string>walldecor</string>
            <string>other</string>
        </array>
    </dict>
    <!— ... —>
```

NEW


My Albums
Search
Library
Photos
Memories
People
Places
Imports
Albums
My Albums
Projects
My Projects
Picnic
May 2017
25 Photos
Show
Showing: All Photos
My Extension
Softcover
A personalized softcover book with high-quality binding and photo cover.
Square
A square hardcover book featuring photo-wrapped cover, matching dust jacket, and protective sleeve.
Classic Book
Your own custom hardcover book, complete with photo-wrapped cover, matching dust jacket, and protective sleeve.
Cancel
Create
MacBook Pro


My Albums
Search
Library
Photos
Memories
People
Places
Imports
Albums
My Albums
Projects
My Projects
Picnic
May 2017
25 Photos
Show
Showing: All Photos
My Extension
Softcover
A personalized softcover book with high-quality binding and photo cover.
From $9.99
Square
A square hardcover book featuring photo-wrapped cover, matching dust jacket, and protective sleeve.
From $24.99
Classic Book
Your own custom hardcover book, complete with photo-wrapped cover, matching dust jacket, and protective sleeve.
From $29.99
Prices do not include taxes and shipping cost. Terms and conditions apply.
Cancel
Create
MacBook Pro

```
// Dynamic Project Type Description Updates

protocol PHProjectExtensionController : NSObjectProtocol {
   func typeDescriptionDataSource(for category: PHProjectCategory,
                                  invalidator: PHProjectTypeDescriptionInvalidator)
       -> PHProjectTypeDescriptionDataSource
    ...
}
protocol PHProjectTypeDescriptionDataSource : NSObjectProtocol {
    func subtypes(for projectType: PHProjectType) -> [PHProjectTypeDescription]
    func footerText(forSubtypesOf projectType: PHProjectType) -> NSAttributedString?
   ...
}
protocol PHProjectTypeDescriptionInvalidator : NSObjectProtocol {
    func invalidateTypeDescription(for projectType: PHProjectType)
    func invalidateFooterText(forSubtypesOf projectType: PHProjectType)
}
```

NEW

```
// Dynamic Project Type Description Updates

protocol PHProjectExtensionController : NSObjectProtocol {
    func typeDescriptionDataSource(for category: PHProjectCategory,
                                   invalidator: PHProjectTypeDescriptionInvalidator)
        -> PHProjectTypeDescriptionDataSource
    ...
}
protocol PHProjectTypeDescriptionDataSource : NSObjectProtocol {
    func subtypes(for projectType: PHProjectType) -> [PHProjectTypeDescription]
    func footerText(forSubtypesOf projectType: PHProjectType) -> NSAttributedString?
    ...
}
protocol PHProjectTypeDescriptionInvalidator : NSObjectProtocol {
    func invalidateTypeDescription(for projectType: PHProjectType)
    func invalidateFooterText(forSubtypesOf projectType: PHProjectType)
}
```

NEW

```
// Dynamic Project Type Description Updates

protocol PHProjectExtensionController : NSObjectProtocol {
   func typeDescriptionDataSource(for category: PHProjectCategory,
                                  invalidator: PHProjectTypeDescriptionInvalidator)
       -> PHProjectTypeDescriptionDataSource
    ...
}
protocol PHProjectTypeDescriptionDataSource : NSObjectProtocol {
    func subtypes(for projectType: PHProjectType) -> [PHProjectTypeDescription]
    func footerText(forSubtypesOf projectType: PHProjectType) -> NSAttributedString?
   ...
}
protocol PHProjectTypeDescriptionInvalidator : NSObjectProtocol {
    func invalidateTypeDescription(for projectType: PHProjectType)
    func invalidateFooterText(forSubtypesOf projectType: PHProjectType)
}
```

NEW

```
// Dynamic Project Type Description Updates

protocol PHProjectExtensionController : NSObjectProtocol {
   func typeDescriptionDataSource(for category: PHProjectCategory,
                                  invalidator: PHProjectTypeDescriptionInvalidator)
       -> PHProjectTypeDescriptionDataSource
    ...
}
protocol PHProjectTypeDescriptionDataSource : NSObjectProtocol {
    func subtypes(for projectType: PHProjectType) -> [PHProjectTypeDescription]
    func footerText(forSubtypesOf projectType: PHProjectType) -> NSAttributedString?
   ...
}
protocol PHProjectTypeDescriptionInvalidator : NSObjectProtocol {
    func invalidateTypeDescription(for projectType: PHProjectType)
    func invalidateFooterText(forSubtypesOf projectType: PHProjectType)
}
```

NEW

PHProjectInfo

```
// PHProjectInfo
var sections: [PHProjectSection]
```

```
// PHProjectSection
var sectionType: PHProjectSection.SectionType
var sectionContents: [PHProjectSectionContent]
```

```
// PHProjectSectionContent
var elements: [PHProjectElement]
var numberOfColumns: Int
var aspectRatio: Double

var cloudAssetIdentifiers: [PHCloudIdentifier]
```

PHProjectInfo

```
// PHProjectInfo
var sections: [PHProjectSection]
```

```
// PHProjectSection
var sectionType: PHProjectSection.SectionType
var sectionContents: [PHProjectSectionContent]
```

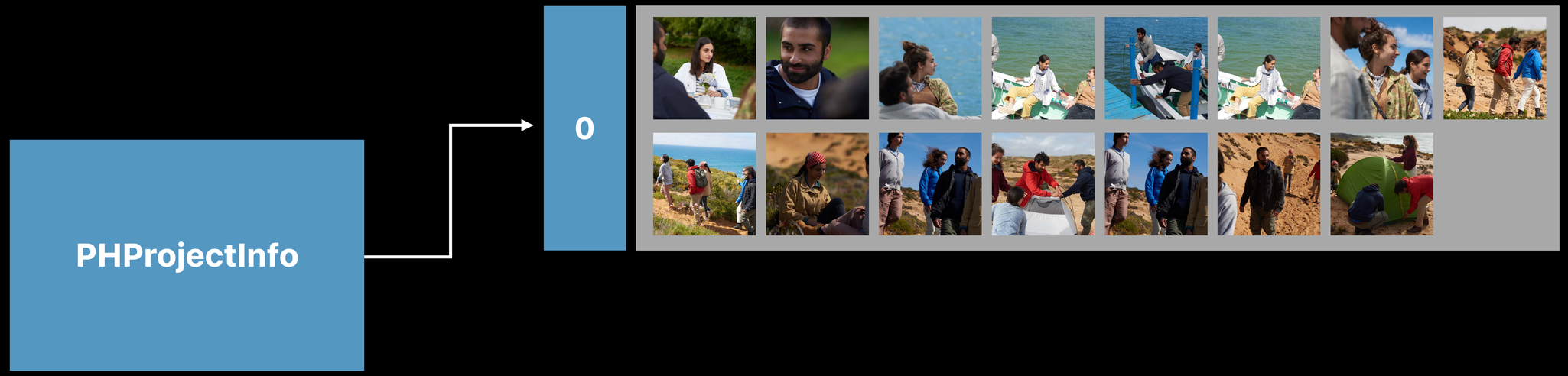
```
// PHProjectSectionContent
var elements: [PHProjectElement]
var numberOfColumns: Int
var aspectRatio: Double

var cloudAssetIdentifiers: [PHCloudIdentifier]
```

PHProjectInfo Updates

Updating appends sections for new assets

NEW

```
projectExtensionContext.updatedProjectInfo(from:completion:)
```



PHProjectInfo Updates

Updating appends sections for new assets

NEW

```
projectExtensionContext.updatedProjectInfo(from:completion:)
```

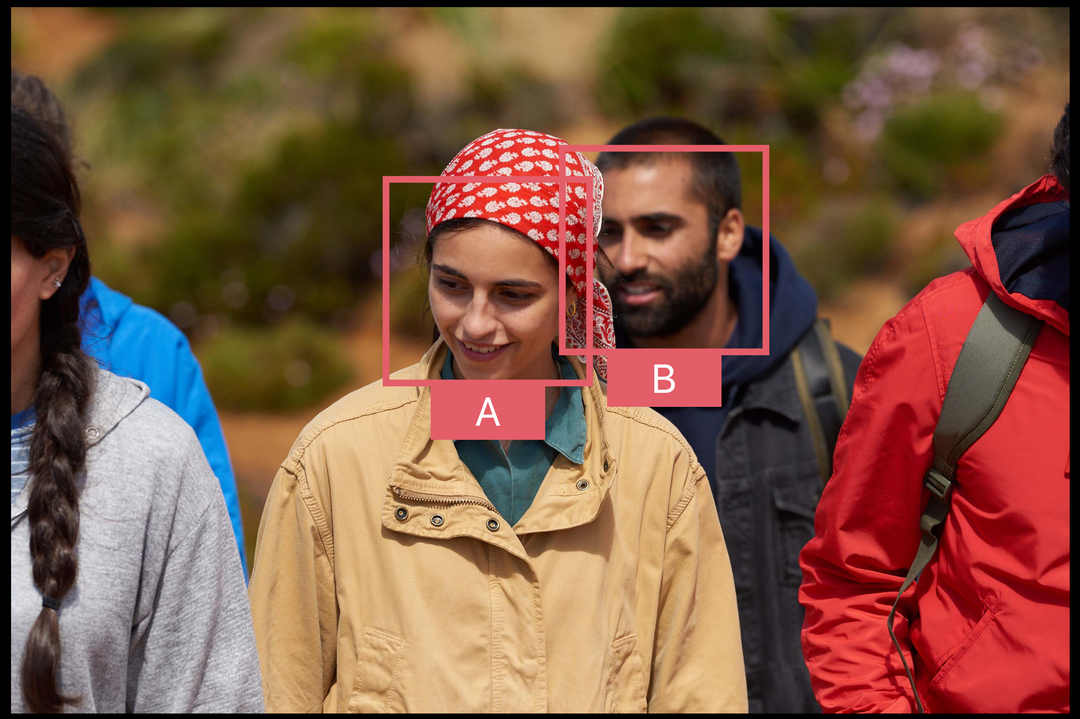
PHProjectInfo Updates

Updating appends sections for new assets

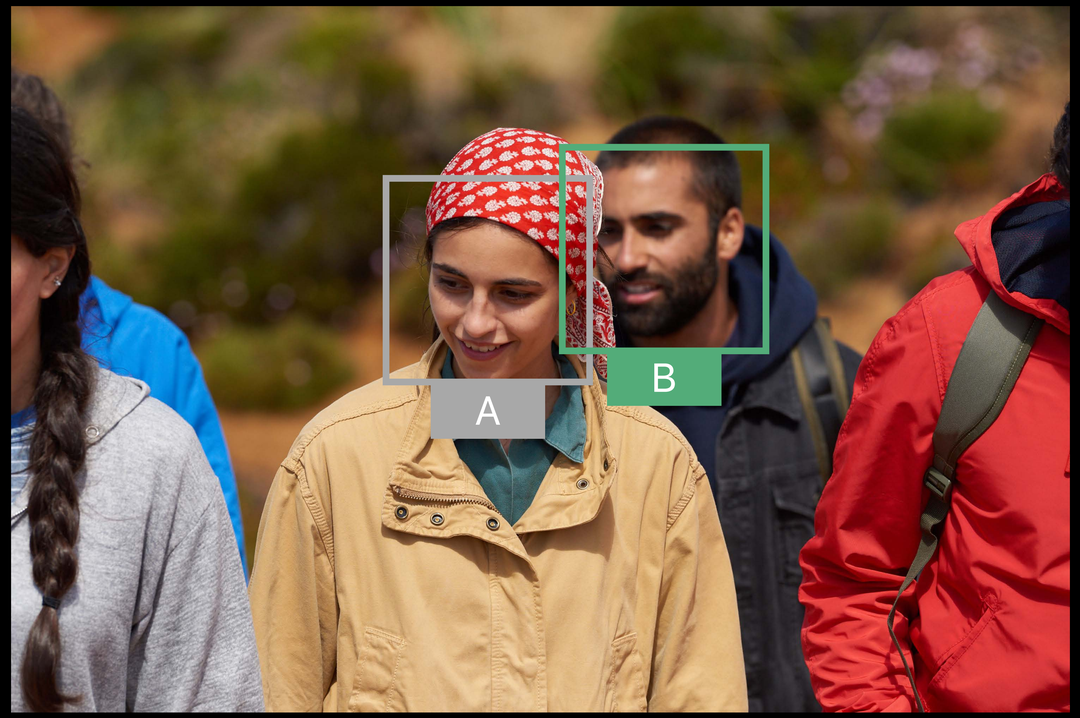
NEW

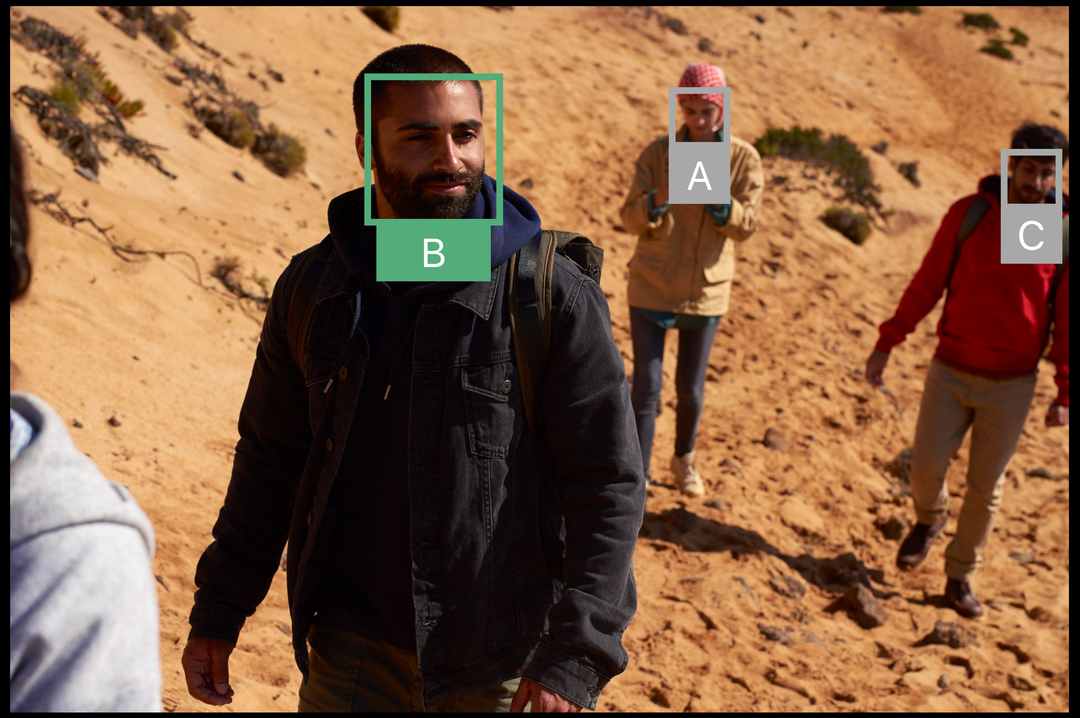
```
projectExtensionContext.updatedProjectInfo(from:completion:)
```

Regions of Interest



Regions of Interest





Regions of Interest

Regions of Interest

NEW

```swift
// PHProjectRegionOfInterest addition


class PHProjectRegionOfInterest : NSObject, NSSecureCoding {

    // Significance of the region of interest (identifier) in the project.
    var weight: Double { get }

    // Quality of the represented region of interest in the asset.
    var quality: Double { get }
}
```

```swift
// PHProjectRegionOfInterest addition


class PHProjectRegionOfInterest : NSObject, NSSecureCoding {

    // Significance of the region of interest (identifier) in the project.
    var weight: Double { get }

    // Quality of the represented region of interest in the asset.
    var quality: Double { get }
}
```

```swift
// PHProjectRegionOfInterest addition


class PHProjectRegionOfInterest : NSObject, NSSecureCoding {

    // Significance of the region of interest (identifier) in the project.
    var weight: Double { get }

    // Quality of the represented region of interest in the asset.
    var quality: Double { get }   NEW
}
```

Demo

Slideshow Extension

Supporting Copy and Paste

```
func validateMenuItem(_ menuItem: NSMenuItem) -> Bool {
    var canHandlePaste = false
    if menuItem.action == #selector(paste(_:)) {
        canHandlePaste = canHandleCurrentPasteboardContent()
    }
    return canHandlePaste
}
```

Implement `validateMenuItem(_ menuItem: NSMenuItem)`

Only return `true` if the extension can handle the current pasteboard content

Supporting Copy and Paste

```
func validateMenuItem(_ menuItem: NSMenuItem) -> Bool {
    var canHandlePaste = false
    if menuItem.action == #selector(paste(_:)) {
        canHandlePaste = canHandleCurrentPasteboardContent()
    }
    return canHandlePaste
}
```

Implement `validateMenuItem(_ menuItem: NSMenuItem)`

Only return `true` if the extension can handle the current pasteboard content

Supporting Copy and Paste

```
func validateMenuItem(_ menuItem: NSMenuItem) -> Bool {
    var canHandlePaste = false
    if menuItem.action == #selector(paste(_:)) {
        canHandlePaste = canHandleCurrentPasteboardContent()
    }
    return canHandlePaste
}
```

Implement `validateMenuItem(_ menuItem: NSMenuItem)`

Only return `true` if the extension can handle the current pasteboard content

Supporting Drag and Drop

Photos handles drags of assets to the extension by default

The extension should only handle extension-internal drags

Interacting with Photos

Dragging items between applications

Sanaa Squalli, Photos Engineer
Joachim Fornallaz, Photos Engineer

Photos | Moments | Collections | Years

Search

Library
- Photos
- Memories
- People
- Places
- Imports

Albums
- My Albums

Projects
- My Projects

May 2017

May 8, 2017 >

Showing: All Photos

May 8, 2017 >

May 9, 2017 >

New Note

June 3, 2018 at 9:43 AM

Photos | Moments | Collections | Years

Search

Library
- Photos
- Memories
- People
- Places
- Imports

Albums
- My Albums

Projects
- My Projects

May 2017

May 8, 2017 >

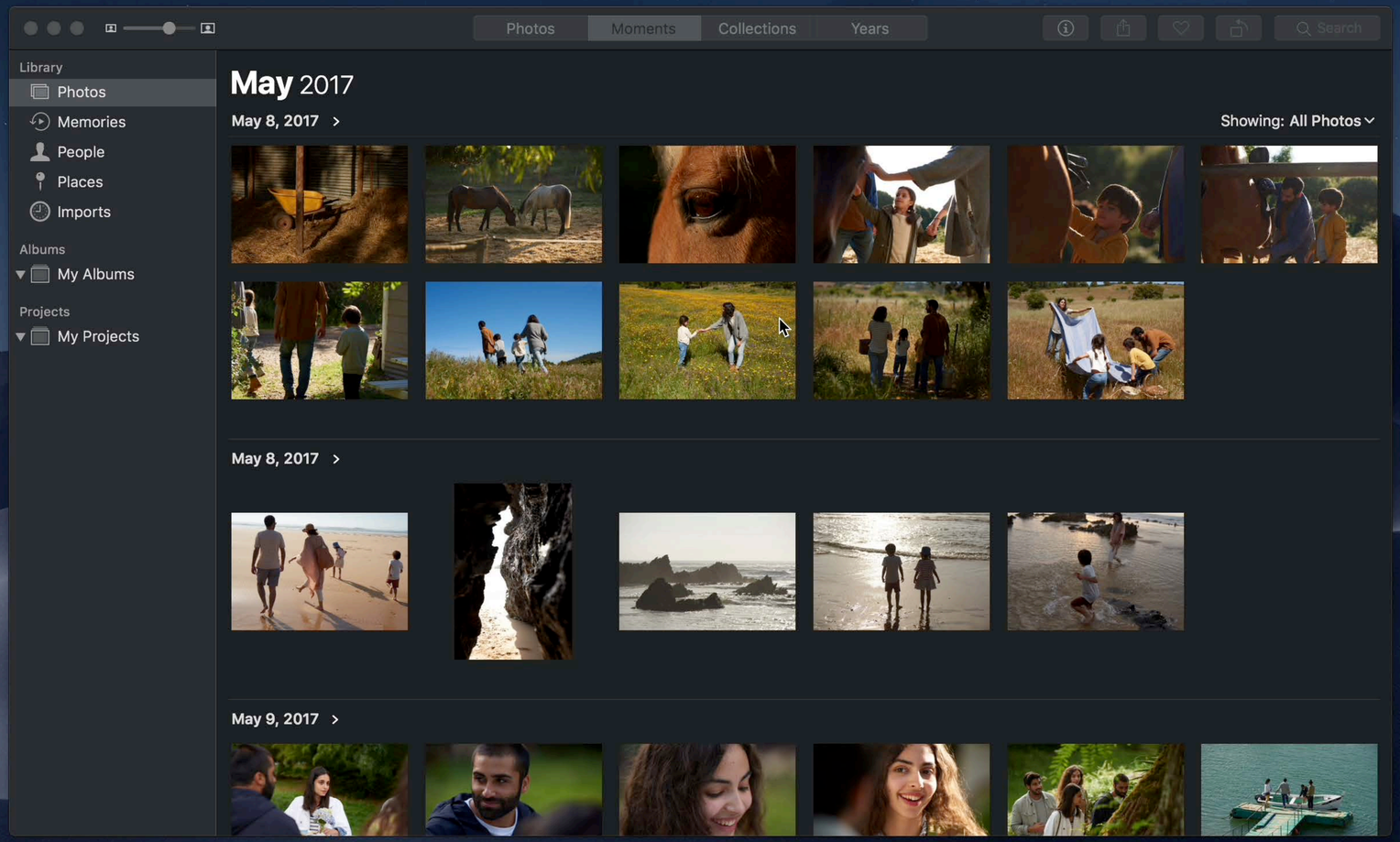
Showing: All Photos

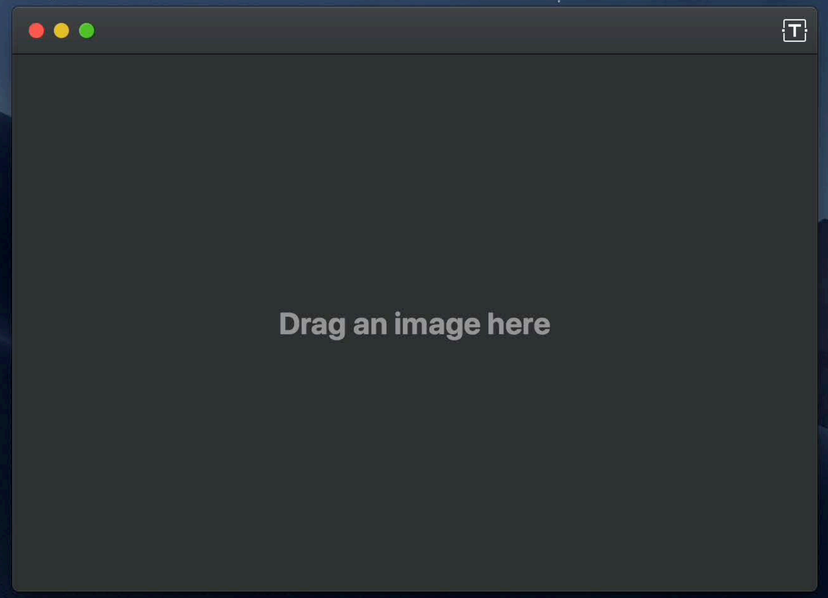
May 8, 2017 >

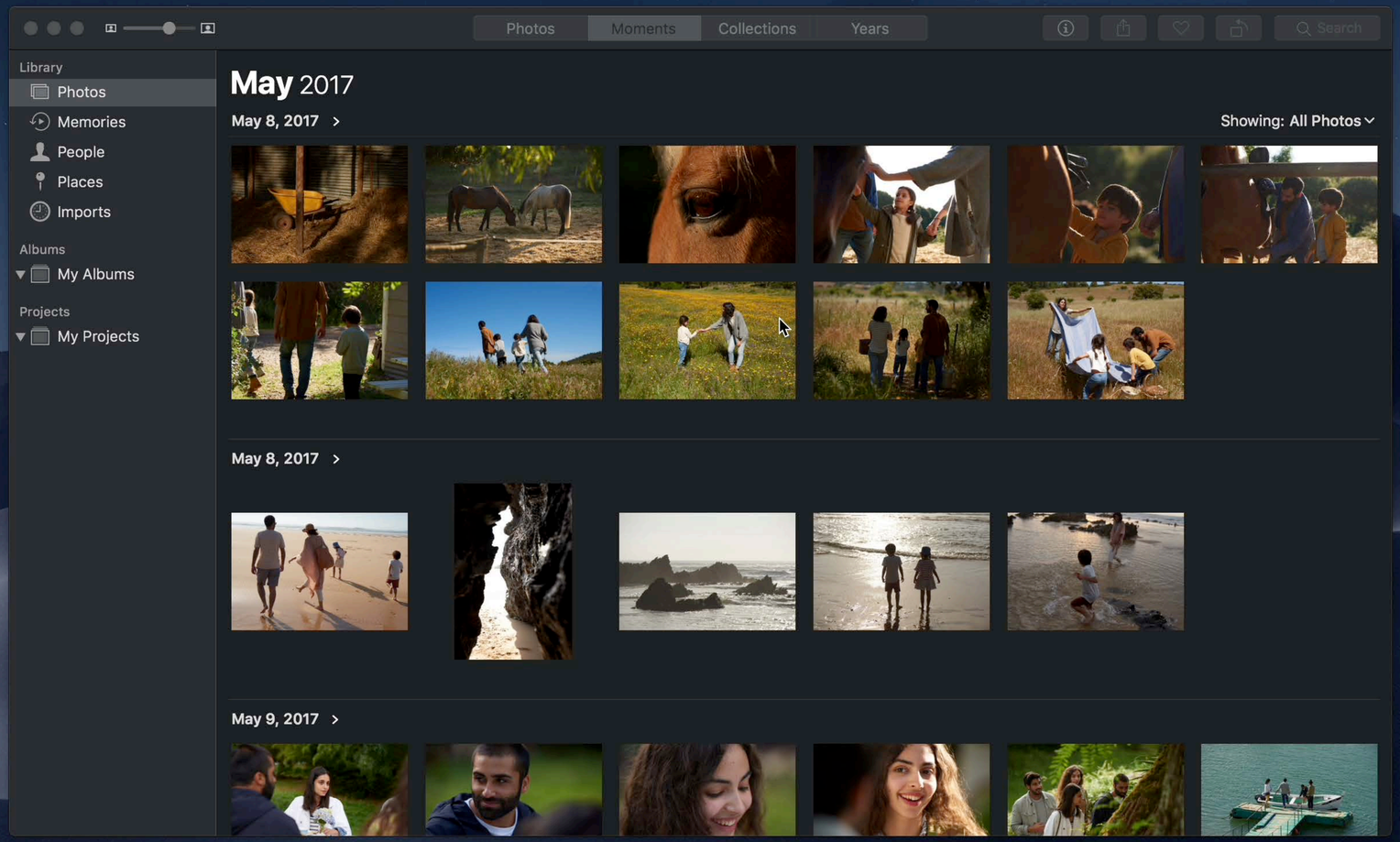
May 9, 2017 >

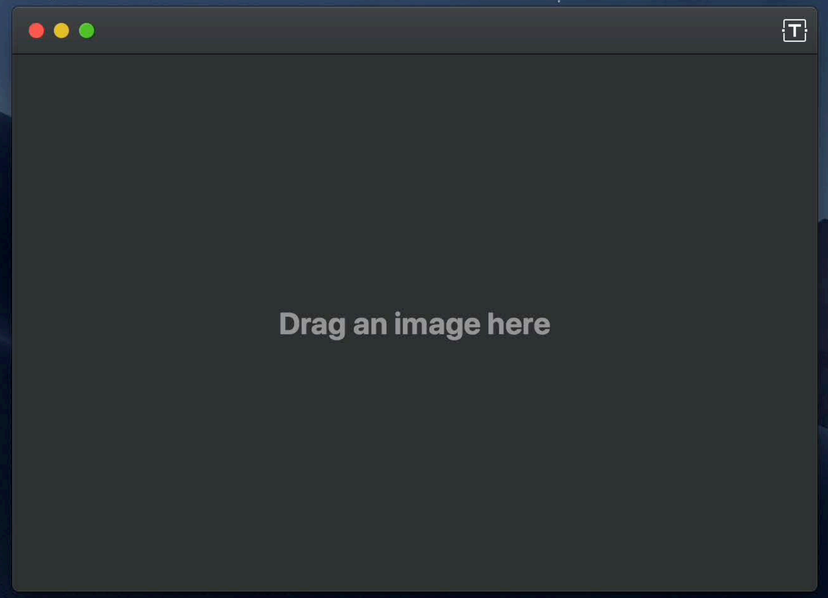
New Note

June 3, 2018 at 9:43 AM


Photos
Moments
Collections
Years
Search
Library
Photos
Memories
People
Places
Imports
Albums
My Albums
Projects
My Projects
May 2017
May 8, 2017
Showing: All Photos
May 8, 2017
May 9, 2017


Drag an image here


Photos
Moments
Collections
Years
Search
Library
Photos
Memories
People
Places
Imports
Albums
My Albums
Projects
My Projects
May 2017
May 8, 2017
Showing: All Photos
May 8, 2017
May 9, 2017


Drag an image here

Drag and Drop on macOS

Uses `NSPasteboard`

Data is written and read using the main thread

Then and Now

Only local content

Read and write file URLs to the pasteboard

Then and Now

Data in iCloud with Optimized Local Storage

Privacy Settings

Metadata: Include location information for published items

Need for Asynchronous API

File Promises

File Promises

What is a file promise?

- Promise that a file of a certain type will be written to a provided location
- Allows the sender to write a promised file in the background

File Promises

What is a file promise?

- Promise that a file of a certain type will be written to a provided location
- Allows the sender to write a promised file in the background

- `NSFilePromiseReceiver`
- `NSFilePromiseProvider`

Receiving File Promises

Always accept both
file URLs and file promises.

File Promise Providers

Photos

Mail

Safari

Keynote

File Promise Providers

Photos

Mail

Safari

Keynote

File Promise Providers

Photos

Mail

Safari

Keynote

File Promise Providers

Photos

Mail

Safari

Keynote

File Promise Providers

Photos

Mail

Safari

Keynote

File Promise Providers

Photos

Mail

Safari

Keynote

```swift
// Reading File Promises

// Register for file promise types during setup
view.registerForDraggedTypes(NSFilePromiseReceiver.readableDraggedTypes.map {
    NSPasteboard.PasteboardType($0)
})
```

```swift
// Reading File Promises

// Register for file promise types during setup
view.registerForDraggedTypes(NSFilePromiseReceiver.readableDraggedTypes.map {
    NSPasteboard.PasteboardType($0)
})
```

```swift
// Reading File Promises

let supportedClasses = [
    NSFilePromiseReceiver.self, NSURL.self, NSImage.self
]
draggingInfo.enumerateDraggingItems(options: [], for: nil, classes: supportedClasses,
                                    searchOptions: options) { (draggingItem, _, _) in
    switch draggingItem.item {
    case let filePromiseReceiver as NSFilePromiseReceiver:
        // Handle file promise
    case let fileURL as URL:
        // Handle file URL
    case let image as NSImage:
        // Handle image
    default: break
    }
}
```

```swift
// Reading File Promises

let supportedClasses = [
    NSFilePromiseReceiver.self, NSURL.self, NSImage.self
]
draggingInfo.enumerateDraggingItems(options: [], for: nil, classes: supportedClasses,
                                    searchOptions: options) { (draggingItem, _, _) in
    switch draggingItem.item {
    case let filePromiseReceiver as NSFilePromiseReceiver:
        // Handle file promise
    case let fileURL as URL:
        // Handle file URL
    case let image as NSImage:
        // Handle image
    default: break
    }
}
```

```swift
// Reading File Promises

let supportedClasses = [
    NSFilePromiseReceiver.self, NSURL.self, NSImage.self
]
draggingInfo.enumerateDraggingItems(options: [], for: nil, classes: supportedClasses,
                                    searchOptions: options) { (draggingItem, _, _) in
    switch draggingItem.item {
    case let filePromiseReceiver as NSFilePromiseReceiver:
        // Handle file promise
    case let fileURL as URL:
        // Handle file URL
    case let image as NSImage:
        // Handle image
    default: break
    }
}
```

```swift
// Reading File Promises

let supportedClasses = [
    NSFilePromiseReceiver.self, NSURL.self, NSImage.self
]
draggingInfo.enumerateDraggingItems(options: [], for: nil, classes: supportedClasses,
                                    searchOptions: options) { (draggingItem, _, _) in
    switch draggingItem.item {
    case let filePromiseReceiver as NSFilePromiseReceiver:
        // Handle file promise
    case let fileURL as URL:
        // Handle file URL
    case let image as NSImage:
        // Handle image
    default: break
    }
}
```

```
// Reading File Promises

draggingInfo.enumerateDraggingItems(options: [], for: nil, classes: supportedClasses,
                                    searchOptions: options) { (draggingItem, _, _) in
    switch draggingItem.item {
    case let filePromiseReceiver as NSFilePromiseReceiver:
        filePromiseReceiver.receivePromisedFiles(atDestination: url, options: [:],
                                                 operationQueue: queue) { (fileURL, error) in
            // Handle file
        }
    case let fileURL as URL:
        // Handle file URL
    case let image as NSImage:
        // Handle image
    default: break
    }
}
```

```
// Reading File Promises

draggingInfo.enumerateDraggingItems(options: [], for: nil, classes: supportedClasses,
                                    searchOptions: options) { (draggingItem, _, _) in
    switch draggingItem.item {
    case let filePromiseReceiver as NSFilePromiseReceiver:
        filePromiseReceiver.receivePromisedFiles(atDestination: url, options: [:],
                                                 operationQueue: queue) { (fileURL, error) in
            // Handle file
        }
    case let fileURL as URL:
        // Handle file URL
    case let image as NSImage:
        // Handle image
    default: break
    }
}
```

```
// Reading File Promises

draggingInfo.enumerateDraggingItems(options: [], for: nil, classes: supportedClasses,
                                    searchOptions: options) { (draggingItem, _, _) in
    switch draggingItem.item {
    case let filePromiseReceiver as NSFilePromiseReceiver:
        // Display file preview
        filePromiseReceiver.receivePromisedFiles(atDestination: url, options: [:],
                                                 operationQueue: queue) { (fileURL, error) in
            // Handle file
        }
    case let fileURL as URL:
        // Handle file URL
    case let image as NSImage:
        // Handle image
    default: break
    }
}
```

```
// Reading File Promises

draggingInfo.enumerateDraggingItems(options: [], for: nil, classes: supportedClasses,
                                    searchOptions: options) { (draggingItem, _, _) in
    switch draggingItem.item {
    case let filePromiseReceiver as NSFilePromiseReceiver:
        // Display file preview
        filePromiseReceiver.receivePromisedFiles(atDestination: url, options: [:],
                                                 operationQueue: queue) { (fileURL, error) in
            // Load file and display content
        }
    case let fileURL as URL:
        // Handle file URL
    case let image as NSImage:
        // Handle image
    default: break
    }
}
```

Photos | Moments | Collections | Years | Search

Library
- Photos
- Memories
- People
- Places
- Imports
- Recently Deleted

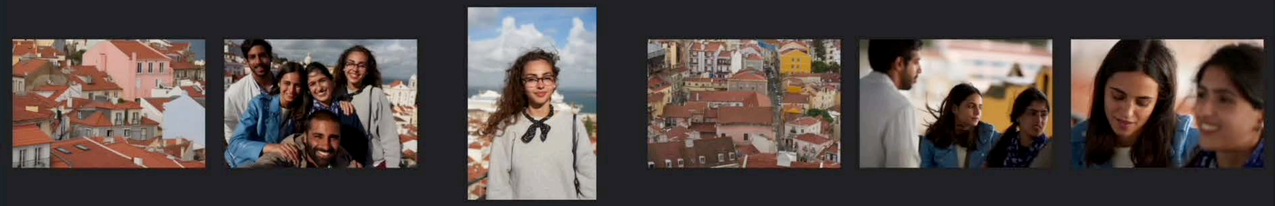
Albums
- My Albums

Projects
- My Projects

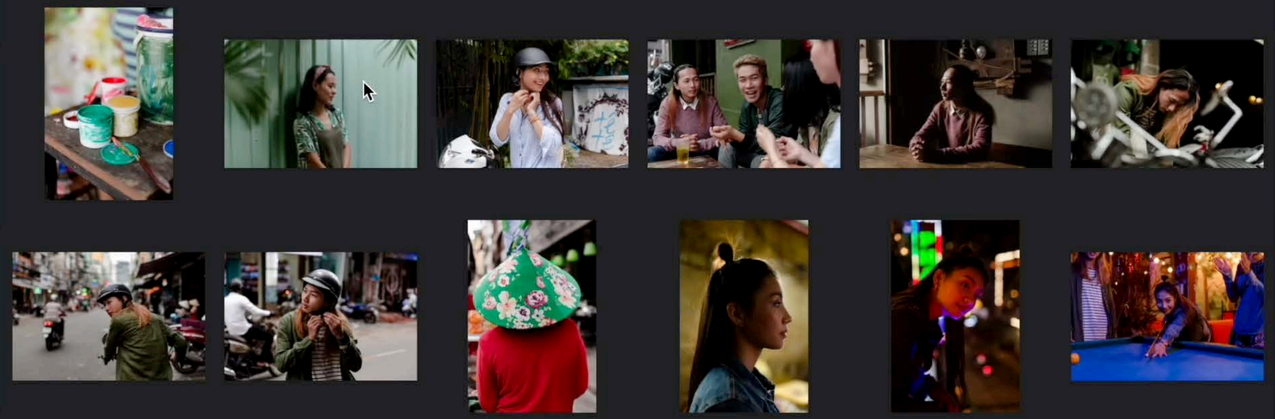
May – June 2017

May 11, 2017 >

Showing: All Photos



Jun 11, 2017 >



To:

Cc:

Subject:

Photos | Moments | Collections | Years | Search

Library
- Photos
- Memories
- People
- Places
- Imports
- Recently Deleted

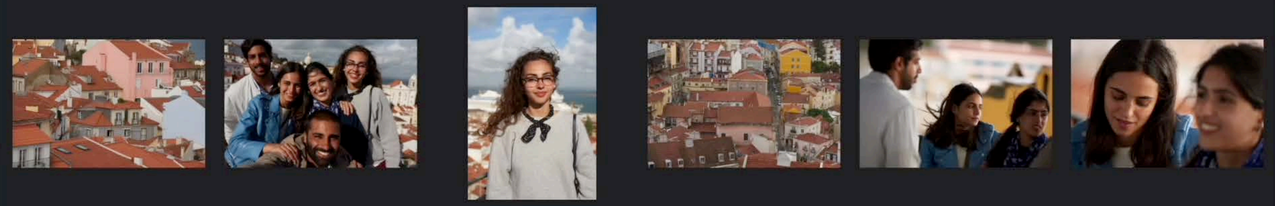
Albums
- My Albums

Projects
- My Projects

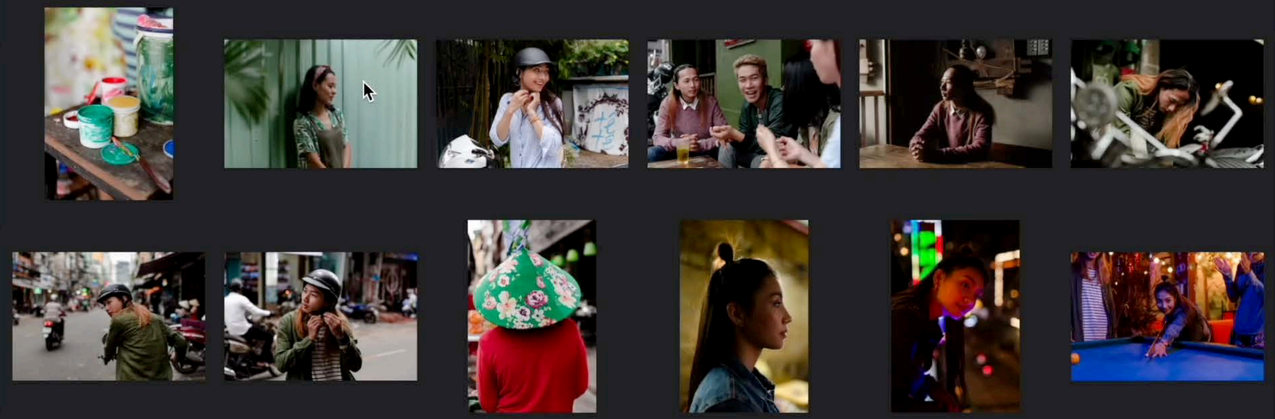
May – June 2017

May 11, 2017 >

Showing: All Photos



Jun 11, 2017 >



To:

Cc:

Subject:

Providing File Promises

...when the data isn’t available on disk.

```swift
// Providing File Promises

// 1. Create file promise provider
let provider = NSFilePromiseProvider(fileType: kUTTypeJPEG as String, delegate: self)

// 2. Implement NSFilePromiseProviderDelegate methods
func filePromiseProvider(_ filePromiseProvider: NSFilePromiseProvider,
                         fileNameForType fileType: String) -> String {
    return "WWDC18.jpg"
}

func operationQueue(for filePromiseProvider: NSFilePromiseProvider) -> OperationQueue {
    return providerQueue
}

func filePromiseProvider(_ promiseProvider: NSFilePromiseProvider, writePromiseTo url: URL,
                         completionHandler: @escaping (Error?) -> Void) {
    // Write file to url
    completionHandler(error)
}
```

```swift
// Providing File Promises

// 1. Create file promise provider
let provider = NSFilePromiseProvider(fileType: kUTTypeJPEG as String, delegate: self)

// 2. Implement NSFilePromiseProviderDelegate methods
func filePromiseProvider(_ filePromiseProvider: NSFilePromiseProvider,
                         fileNameForType fileType: String) -> String {
    return "WWDC18.jpg"
}

func operationQueue(for filePromiseProvider: NSFilePromiseProvider) -> OperationQueue {
    return providerQueue
}

func filePromiseProvider(_ promiseProvider: NSFilePromiseProvider, writePromiseTo url: URL,
                         completionHandler: @escaping (Error?) -> Void) {
    // Write file to url
    completionHandler(error)
}
```

```swift
// Providing File Promises

// 1. Create file promise provider
let provider = NSFilePromiseProvider(fileType: kUTTypeJPEG as String, delegate: self)

// 2. Implement NSFilePromiseProviderDelegate methods
func filePromiseProvider(_ filePromiseProvider: NSFilePromiseProvider,
                         fileNameForType fileType: String) -> String {
    return "WWDC18.jpg"
}

func operationQueue(for filePromiseProvider: NSFilePromiseProvider) -> OperationQueue {
    return providerQueue
}

func filePromiseProvider(_ promiseProvider: NSFilePromiseProvider, writePromiseTo url: URL,
                         completionHandler: @escaping (Error?) -> Void) {
    // Write file to url
    completionHandler(error)
}
```

```swift
// Providing File Promises

// 1. Create file promise provider
let provider = NSFilePromiseProvider(fileType: kUTTypeJPEG as String, delegate: self)

// 2. Implement NSFilePromiseProviderDelegate methods
func filePromiseProvider(_ filePromiseProvider: NSFilePromiseProvider,
                         fileNameForType fileType: String) -> String {
    return "WWDC18.jpg"
}

func operationQueue(for filePromiseProvider: NSFilePromiseProvider) -> OperationQueue {
    return providerQueue
}

func filePromiseProvider(_ promiseProvider: NSFilePromiseProvider, writePromiseTo url: URL,
                         completionHandler: @escaping (Error?) -> Void) {
    // Write file to url
    completionHandler(error)
}
```

```swift
// Providing File Promises

// 1. Create file promise provider
let provider = NSFilePromiseProvider(fileType: kUTTypeJPEG as String, delegate: self)

// 2. Implement NSFilePromiseProviderDelegate methods
func filePromiseProvider(_ filePromiseProvider: NSFilePromiseProvider,
                         fileNameForType fileType: String) -> String {
    return "WWDC18.jpg"
}

func operationQueue(for filePromiseProvider: NSFilePromiseProvider) -> OperationQueue {
    return providerQueue
}

func filePromiseProvider(_ promiseProvider: NSFilePromiseProvider, writePromiseTo url: URL,
                         completionHandler: @escaping (Error?) -> Void) {
    // Write file to url
    completionHandler(error)
}
```

```swift
// Providing File Promises

// 1. Create file promise provider
let provider = NSFilePromiseProvider(fileType: kUTTypeJPEG as String, delegate: self)

// 2. Implement NSFilePromiseProviderDelegate methods
func filePromiseProvider(_ filePromiseProvider: NSFilePromiseProvider,
                         fileNameForType fileType: String) -> String {
    return "WWDC18.jpg"
}

func operationQueue(for filePromiseProvider: NSFilePromiseProvider) -> OperationQueue {
    return providerQueue
}
```

```swift
func filePromiseProvider(_ promiseProvider: NSFilePromiseProvider, writePromiseTo url: URL,
                         completionHandler: @escaping (Error?) -> Void) {
    // Write file to url
    completionHandler(error)
}
```

Demo

Adapting your app for File Promises

Joachim Fornallaz, Photos Engineer

Summary

Support File Promises

Build Photo Project Extensions

More Information

https://developer.apple.com/wwdc18/505

PhotoKit and Photos Extensions Lab	Technology Lab 3	Thursday 3:00 PM

WWDC18